计算语言学的理论方法和研究取向

来源：英特网 时间：2005-7-7 9:09:45

袁毓林

【摘 要 题】本文从不同的研究取向的角度，对目前计算语言学的几种理论方法以及相应的语言处理技术进行比较研究。着重讨论工程主义、工具主义、认知主义、实证主义和逻辑主义五种研究取向，比较对人类知识和语言理解过程及相应的计算机模拟策略的几种不同的理论，分析其在具体的语言处理技术（包括语法形式体系、语义表示体系、分析算法以至程序实现）上的差异。希望对不同的理论方法与处理技术的效能和局限有一个比较清楚的认识，从而为汉语计算语言学的研究提供借鉴。
【关 键 词】计算语言学/语法形式体系/语义表示体系/分析算法/程序实现

0.计算机：语言研究的奴仆还是上帝

计算语言学是一门跟当代科学技术关系最密切的学科，同时也是一门定义最为纷歧的学科。只要打开有关的文献，你就能找到关于计算语言学的各种差别极大的定义。事实上，这些不同的定义背后反映了不同的研究者的不同的研究取向。其中，最核心的一点是：怎样看待计算机和语言研究的关系，是把计算机作为语言研究的工具、还是作为语言研究的目标和服务对象。形象地说，把计算机当做为语言研究服务的奴仆、还是当做语言研究要为之服务的上帝。

下面，我们通过五种关于计算语言学的定义，来讨论工程主义、工具主义、认知主义、实证主义和逻辑主义五种不同的研究取向，比较不同的研究者为了实现这些不同的目标而采用的迥然不同的理论和方法（包括对人类知识、语言习得和语言理解过程的看法以及相应的在计算机上模拟的策略），分析其在具体的语言处理技术（包括语法形式体系、语义表示体系、分析算法以至程序实现）上的差异。希望对计算语言学中不同的理论方法与处理技术的效能和局限有一个比较清楚的认识，从而为汉语计算语言学的研究提供借鉴。

1.工程主义取向：着眼于计算机系统的建立

在计算语言学的诸多定义中，最多的是着眼于建立一种可运转的计算机系统。例如：

(1)计算语言学是对能理解和生成自然语言的计算机系统的研究。

——Grishman(1986)§1.1,p.4

(2)计算语言学是采用计算机技术来研究和处理自然语言的一门新兴学科。

——冯志伟(1992)，第84页

持这种观点的学者自然会把计算语言学的研究重点放在这种能理解和生成自然语言的计算机系统的结构及相应的各种算法的设计上。因为，从理论上说，要想让计算机去解决某种问题，必须满足下列三个基本的前提条件：（注：详见马希文《计算机和思维科学》，§2，见钱学森主编《关于思维科学》，人民出版社1986年版，第225-228页。）

第一，必须把待解的问题形式化。由于计算机只能对有限符号集上的有限长度的符号序列进行决定型的形式变换（这就是计算），因而首先要建立一个形式体系（formalism，一译形式系统）：规定所用的各种符号（词汇），规定把符号连接成合法序列（即合式公式）的规则（句法），规定合法的符号串如何表示特定问题领域中的意义（语义，或解释）；然后，建立一些推理规则，说明对这些符号和合法符号串可以进行一些什么样的处理（演算）。于是，问题便可以用符号表达出来，问题的解也表现为对符号序列的条件。这样，计算机解决问题的过程就是从表示问题的符号序列出发，按规则进行加工，一直到得出符号要求的符号序列（即解）为止。这一整

套的办法叫形式化（又叫数学方法），其要义是：把特定领域的问题转变为符号，从而把对问题的求解转变为对符号串的变换处理。

第二，这种问题必须是可计算的(computable)，即一定要有解题的算法(algorithm)，使得计算机能按照算法所指引的解题步骤，通过有限步的运算而得出结果。

第三，这种问题必须有一个合理的复杂度，也就是要避免指数爆炸(exponential explosion)。也就是说，问题的复杂性必须限制在目前的数学计算机的存储空间和运算时间所能容忍的范围之内。

所以，从研究程序上讲，这种类型的计算语言学研究一般分为如下三个阶段：（注：参考冯志伟《计算语言学对理论语言学的挑战》，《语言文字应用》1992年第1期；钱锋《计算语言学引论》，学林出版社1990年版，第26-27页。）

第一步，数学建模。把需要研究的问题在语言学上加以形式化(linguistic formalism)，使之能以一定的数学形式、严密而规整地表示出来。也就是说，为有关的语言问题建立数学模型。包括选择恰当的形式语法(formal grammar)使得句子的结构能够用某种数学形式明确而清晰地表示出来，研究在这种形式语法之下如何分析句子构造的方法和步骤；选择恰当的表示体系使得句子的意义能够用某种数学形式明确而清晰地表示出来，研究在这种形式体系之下如何分析和表示句子的语义结构。

第二步，算法设计。把这种严密而规整的数学形式表示为算法，使之在计算上形式化(computational formalism)。这就必须研究句子分析的严格的手续(procedures)，并抽象成机械的、明确的、一步步逼近分析结果的步骤。

第三步，程序实现。根据算法用某种程序语言编写计算机程序，使之在计算机上加以实现(computer implementation)。

比如，假定有下面这部小型的用产生式(production)表示的语境自由的短语结构语法：

S→NP+VP　　……R1

NP→N　　……R2

NP→PRO　　……R3

VP→Vi　　……R4

VP→Vt+NP　　……R5那么，句子I like cheese.（我喜欢奶酪）的最左推导是：

S →NP+VP

→PRO+VP

→PRO+Vt+NP

→PRO+Vt+N

为了让计算机能根据上面给出的语法规则自动地分析这个句子，必须设计相应的算法：或者是自顶向下的回溯算法，或者是自底向上的并行算法。自顶向下的回溯算法每次只尝试一种推导，当一种推导失败时便返回、重新尝试另一种推导；就这样逐个地枚举语法所允许的各种推导，直至找到一个能生成输入句子的推导。根据这种算法（具体的细节从略），对于上文那部只有五条规则的语法，句子I like cheese.的推导过程将表现如下：

i.S

ii.S→NP+VP

iii.S→NP+VP→N+VP

iv.S→NP+VP→PRO+VP

v.S→NP+VP→PRO+VP→PRO+Vi

vi.S→NP+VP→PRO+VP→PRO+Vt+NP

vii.S→NP+VP→PRO+VP→PRO+Vt+N

i.S是初始符，即树顶节点；ii.根据R1，展开初始符；iii.根据R2展开最左的非终结符，但是范畴N跟词项I不匹配，需要回溯；iv.根据R3展开最左的非终结符，范畴TPO跟词项I匹配成功；v.根据R4展开左端第二个非终结符，但是范畴Vi跟词项like不匹配，需要回溯；vi.根据R5展开左端第二个非终结符，范畴Vt跟词项like匹配成功；vii.根据R2展开最后一个非终结符，范畴N跟词项cheese匹配成功；至此，推导结束。（注：详见石纯一等《人工智能原理》§9.4.2，清华大学出版社1993年版，第355-363页。）

一般地说，计算语言学的研究必须涉及计算机科学中的复杂性理论（complexity theory，用以判别所研究的问题是否具有可计算性）、编译技术(compiler technology)、搜索策略(search strategies)、真值保持系统(truth-maintenance systems)、自动定理证明(automatic theorem proving)、知识表示(knowledge representation)和数据结构(datastructure)等方面，同时也必须涉及语言学中的语音学(phonetics)、音系学(phonology)、形态学（morphology，或词法学）、句法学(syntax)、语义学(semantics)、语用学

(pragmatics)、话语分析(discourse analysis)等方面。见右图表。（注：参考Halvorsen(1988)§3:Computer applications of linguistic theory,pp.202-203。）

附图 H11I 01.JPG

如果说科学是理论和知识体系、技术是方法和操作技巧、工程是实践和具体施行的话，那么计算语言学就是一种工程。为了建造一个顺畅(fluent)、健壮(robust)的自然语言处理系统，必须整合许多不同类型的知识，诸如句法知识、语义知识、话语领域知识等，并且要有效地用到自然语言处理系统中。正是在这一意义上，建造处理自然语言的计算机系统跟建造其他大型的计算机系统一样，主要是一种工程性的工作。跟其他系统建造工作一样，计算语言学采用模块化(modularity)和建立形式模型(formal models)两种通用技术。所谓模块化，是指把我们的系统所涉及的知识分割为相对独立的成分，然后分别攻克一个个子问题，从而缩小整个系统的规模。所谓建立形式模型，是指为复杂系统建立一种相对简单的抽象模型，然后为这种简化的模型设计我们的计算机系统。（注：详见Grishman(1986)§1.3:Computational linguistics as engineering,pp.7-8。）

这种工程主义取向的计算语言学研究是有很强的应用动机的。因为语言是人类交际和记录信息的工具(vehicle)，如果使计算机获得生成和理解自然语言的能力，那么计算机就能执行只有人类才能完成的工作，诸如翻译、文本处理、信息抽取和检索等；所以，能处理自然语言的计算机系统将使计算机更为有用。（注：详见Grishman(1986)Introduction.p.1。）也就是说，通过计算语言学的研究，可以开发更多的计算机应用领域。

2.工具主义取向：着眼于用计算机作语言分析

计算语言学最宽泛的定义是：用计算机来研究和处理自然语言。例如：

(1)计算语言学是关于自然语言的计算机处理的一门学科。它用计算机技术来研究和处理自然语言。——陆致极(1990)，第15页

(2)对计算语言学一般有狭义的和广义的两种理解。狭义理解盛行于计算语言学最为发达的美国，它大致上就是人工智能中自然语言理解（包括机器翻译）的理论和方法部门，它的操作内容大致上就是上面所提到的(1)-(5)。（注：这里的(1)-(5)就是§1中第一步至第三步的内容。）广义的理解则把凡是利用计算机处理自然语言的有关问题（例如，……风格研究）都囊括进来了，这种理解欧洲比较盛行。——钱锋(1990)，第27-28页

在这种包容性很大的定义中，除了有§1中讨论的研究能理解自然语言的计算机系统之外，还有利用计算机来进行跟语言相关的研究等内容，比如，用计算机对字母频率、汉字频率、词长、句长、句型等语言成分的统计研究，以及建立在语言成分的统计基础上的作品风格研究和匿名作品的作者考证研究等。简单地说，工具主义取向的计算语言学着眼于用计算机来进行语言的计量研究(quantitative studies)。

值得一提的是，随着用计算机来采集、整理、加工和管理语言材料工作的深入开展，逐步形成了语料库语言学(corpus linguistics)这门计算语言学的分支学科。大概地说，语料库语言学研究机器可读的(machine-readable)自然语言文本的采集、存储、检索、统计、语法标注(grammatical tagging)、句法语义分析，以及具有上述功能的语料库在语言定量分析、作品风格和作者考证研究、词典编纂、自然语言理解和机器翻译等领域中的运用。比如，为了研究现代美国英语，美国的布朗大学在1964年建立了库容量为100万词的Brown语料库。为了研究现代英国英语，英国的兰开斯特大学跟挪威的奥斯陆大学、卑尔根大学在70年代合作建成LOB语料库，库容量也是100万词。欧美各国的学者利用这两个语料库开展了大规模的英语研究。在1970-1978年间，他们用86种词类标记来对布朗语料库进行语法标注。Greene和Rubin还设计了名叫TAGGIT的自动标注系统，其庞大的规则库里有3300条上下文有关规则。TAGGIT系统对布朗语料库的全部100万词语料进行自动标注的正确率达77%，剩下的同形和兼类歧义问题最后由人工来解决。（注：参考黄昌宁《语料库语言学》，《中国计算机用户》1990年第11期；冯志伟《计算语言学对理论语言学的挑战》，《语言文字应用》1992年第1期。）

从方法论上看，语料库语言学跟工程主义的计算语言学很不相同。后者采用的是以知识（表示成规则）为基础的方法，即人工智能的方法。这种方法假定：如果计算机要处理自然语言，那么它必须跟人一样具有句法、语义、语用、话语篇章、主题事物、周围世界等方面的知识和逻辑推理能力。因为人处理语言时的心理状态和心理过程就是这样的，计算机必须具有跟人相同和相近的知识才能处理自然语言。而语料库语言学采用的则是以语料统计为基础的方法，即基于概率的方法。这种方法认为：计算机并不能像人一样利用知识去理解语言，人们也无法把理解语言所需的各种知识形式化地表示成规则。有鉴于此，这种方法假定：如果我们能对数量很大的语言数据作出定量化的统计分析，那么我们就能对语言成分的分布和语言成分之间的关系等进行概率性的预测，从而补偿计算机缺乏知识和推理能力的缺点。（注：参考桂诗春、宁春岩《语言学方法论》§7.7.2.2：语料库方法，外语教学与研究出版社1997年版，第138-149页。）比如，在1978-1983年间，英国的Leech、Sampson、Garside等人对LOB语料库进行词类标注实验。为此，他们还设计了一个名叫CLAWS的系统(Constituent-Likelihood Automatic Word-tagging System)。他们完全放弃了传统的规则模型，把自动标注的算法建立在统计信息的基础

上。他们采用了133种词类标记，利用已带有语法标记的Brown语料库来获取两个相邻标记的同现频率，据此建立了一个规模为133×133的“标记转移概率矩阵”(tagging transition probability matrix)，用以反映在前一种标记的条件下后一种标记出现的概率。整个语法标注过程所依据的知识都是由这个矩阵提供的。CLAWS系统对LOB语料库的全部100万词语料进行自动标注的正确率达96%，比以规则为基础的TAGGIT系统提高了将近20%。（注：参考黄昌宁(1990)，第44页；桂诗春、宁春岩(1997)，第145页。）例如，对于句子Henry likes stews.，其中Henry是名词短语，只有NP一种标记；likes和stews可以是名词复数或动词第三人称单数，因而有NNS和VBZ两种标记。于是，这三个词可以有下列四种词类搭配方式：

i.NP+NNS+NNS=17×5×135=11475

ii.NP+NNS+VBZ=17×1×37=629

iii.NP+VBZ+NNS=7×28×135=26460

iv.NP+VBZ+VBZ=7×0×37=0

在这些由形式类表示的搭配方式的右侧（等号后面）给出每种标记跟相邻标记的同现概率，并用这种概率的乘积作为决定某种搭配方式的概率的变量。假定决定某种搭配方式的概率等于该变量除于所有变量的和，那么第三种搭配的概率最高(26460/11475+629+26460+0=69%)。系统可以据此确定句子Henry likes stews.的形式类标记是NP+VBZ+NNS。（注：参考桂诗春、宁春岩(1997)§7.7.2.2：语料库方法，第138-149页。）既然通过概率计算可以确定兼类词在某种组合中的词类属性，那么由兼类词引起的结构歧义也可以通过概率计算来消歧（disambiguation或ambiguity resolution）。于是，以语料库为基础的统计模型不仅可以用来解决自然语言的语法标注任务，而且还可以运用到句法、语义等更高层次的分析上来。（注：参考黄昌宁(1990)，第44页。）

3.认知主义取向：着眼于人类使用语言时的心理过程

在计算语言学的定义中，为数不多的是涉及人类使用语言时的心理过程。例如：

(1)计算语言学是一门计算机科学和语言学紧密结合的科学。它用数学的方法来制订语言规则和模型去解决有关计算机的语言学习和理解、语言信息的存储、组织、更新、转换和生成等问题。在这些问题中，核心是学习和理解。——黄建烁(1991)，第24页

(2)计算语言学最好看作是人工智能的一个分支。跟人工智能的所有其他领域一样，它涉及对认知能力的研究和建模。在计算语言学这里，它着重的是语言能力。但是，这种研究不必去建构关于人类行为的具有心理真实性的模型。其目的就在于确定和刻画用自然语言进行交际和获取信息的能力中所包含的知识的种类及相关处理过程的类别，而不管其实际的心理状态。——Halvorsen(1988)§3,p.202

黄建烁(1991)的定义为计算语言学确立了一种非常宏伟的目标，那就是教会机器自动地学习，即让机器理解语言并自动地学习和更新知识。用Hans Karlgreen教授的话来说，就是“用计算的方法来制定人类语言行为的模型，并以此去了解人们怎样听说读写、怎样学习新知识和更新旧知识，又是怎样理解、存储和组织语言信息的”。他甚至认为，计算语言学的一个最根本的问题就是了解“人类的大部分活动在什么程度上能够简化成机械的操作”（注：详见黄建烁《计算语言学研究综述》，《国际学术动态》1991年第4期。）。Halvorsen(1988)则强调，计算语言学是对人类语言处理能力和心理过程的功能（而不是结构）模拟。这就是典型的人工智能方法。这种功能模拟的方法直接影响和促成了认知心理学的基本信念：可以把计算机作为人类思维的模型，也可以用计算机来模拟人类的认知过程。

T.Winograd(1983)Language as a Cognitive Process（〔把〕语言作为一种认知过程〔看待〕），则可以说是认知主义取向的杰出典范。他由下列两个问题激发灵感，尝试建立一种语言研究的认知范式(cognitive paradigm)：

i.一个人要说话和理解语言，必须具有哪些知识？

ii.为了在交际中使用这些知识，人的心智是怎样组织的？

他把语言使用看做是一种以知识为基础的交际过程，认为人无论是说话还是听话都必须具有一定的知识，比如，词序规则、词汇和词的结构、语义特征、所指关系、时制系统、话语结构、说话人的态度、韵律规约、风格规约、世界知识等。在理论方面，他企图探讨人是怎样习得、运用这些知识的；在实际运用方面，他尝试用计算机来模拟人习得、储存、运用这些知识的过程，所以他又称这种范式为计算的范式(computational paradigm)。（注：详见Winograd(1983)chap.1:Viewing Language as a Knowledge-Based Process,pp.1-34。另外，参考黄奕《认知过程的语言》对该书的介绍和评论，《国外语言学》1985年第3期。）

持这种研究取向的学者喜欢用认知心理学的眼光来看待语言使用。从信息加工过程的观点看，人说出一句话和理解一句话时，在大脑中有一个关于所描述的外部世界中的事物或事件的心理映象，可以称之为内部语言；而人处理语言的过程就是把外部语言转化为内部语言，经过加工后再由内部语言转化为外部语言的过程。计算机也可以用类似的过程来处理自然语言：首先确定一种语言的内部表示；然后，寻求一种把所限定的语言子集中的语

句转换为内部表示的方法。在他们看来，让计算机理解语言的关键是：应能对一般的自然语言的句子作出语义解释，即设计一种一般的内部表示。内部表示是自然语言处理的关键，它影响着系统对语言知识和世界知识的描述和利用，因此也影响着整个处理系统。（注：详见杨抒《自然语言的认知模型》，《计算机科学》1988年第3期。）

不同的学者由于对人类处理语言的心理过程的认识不同，因而采用了不同的理论和方法来建造自然语言处理系统。其中一类系统比较重视句法分析，尽管所依据的语法理论各不相同。比如，Winograd 1972年研制的关于积木世界的SHRDLU系统，采用Halliday（1967、1970）的系统语法(Systemic Grammar)，把句法结构看做是生成句子的过程中一系列句法结构选择的结果。Woods1972年设计了关于月球化学成分的L NAR系统，该系统的句法部分根据Chomsky(1965)的转换生成语法，分析出标准理论所指定的深层结构，再输入语义部分。语义部分根据句法上的深层结构再进行语义信息的分析。数据检索部分再根据输入句的语义编译成一种面向系统的形式语言（即查询语句），以便直接查询数据库，并最终产生结果（即回答）。Simmon(1973)根据Fillmore(1968)的格语法(Case Grammar)建立了语义网络理论。他采用Woods的ATN(augmented tran-sition network)来分解输入句的句法关系，同时分析深层格结构，记录语义关系；最后求出输入句的语义关系，据此来理解语义。另一类系统不作详细的句法分析，直接从语句中抽取语义信息。比如，Yorick A.Wilks认为，整段言谈的内容是由一些简单的基本信息构成的。一个复杂的句子也是由基本信息通过概念连结成实时的线性序列，而不是语言学家所认为的具有层次的树形结构。在这种思想的指导下，Wilks(1973)用人工智能的方法设计了一个英法机器翻译的模型。Roger C.Schank认为人脑中存在着某种概念基础(conceptual base)，语言理解的过程就是把语句映射到概念基础上去的过程。概念基础具有完善的结构，人往往能根据初始的输入预期可能的后续信息。句法分析对语言理解的用处不大，因为语言理解需要的是输入句的意思，而不是它的句法结构。计算机要理解语言，必须模拟人的心理过程；要像人一样根据上下文、环境、知识、记忆等作出预期(expectation)，从而获取语义。句法只起一种指引的作用，即根据某些输入词语形成概念结构，预期它的句法形式，便于查找核实。Schank(1973)提出了概念从属(Conceptual Dependency,CD)理论，建立了MARIE模型。上述这些不同的理论和方法，都是基于研究者对于“人是怎样理解语言的”这一问题的不同见解而发展出来的。也就是说，他们分别用不同的计算范式来实现其认知范式。（注：详见杨抒(1988)，第22-26页；范继淹、徐志敏《自然语言理解的理论和方法》，《国外语言学》1980年第5期。）

4.实证主义取向：着眼于检验语法理论的可靠性

跟§1所述的抱有实用目的的工程主义取向不同，大多数计算语言学研究并不跟某种特定的应用目标相挂钩，而是另有某种科学研究的目标。其中之一就是用计算机来对语言学家提出的各种语言学理论进行检验。比如：

计算语言学的一个自然的功能是对理论语言学家提出的各种语法进行检验。——Grishman(1986)§1.1,p.5.

用计算机来检验某种语法理论或某组语法规则，这对语言学家来说实在是一件既令人兴奋又令人不安的事。兴奋的是语言学的理论和规则居然可以像数学公式一样让计算机去执行，不安的是能顺利通过机器检验的希望是极其渺茫的。Friedman(1971)还真的设计了一个检验转换语法的系统，名叫Friedman"s Transformational Grammar Tester。该系统可以按照转换语法来生成句子，于是语言学家可以用它来检验他们的语法是不是真的只生成合语法的句子。事实上，由于大多数语言学理论的形式框架（包括：移位规则的性质、对转换的限制、语义解释规则的形式，等等）都是有问题的，而且理论语言学的重点并不是建造一种能适应计算测试的实体性的语法；因而就目前来看，作为语言学理论的测试工具，计算机的用处是不大的。（注：详见Grishman(1986)§1.1:The objectives of computationallinguistics,p.5。）

看来，让计算语言学来充当语言学理论的审判官是不合适的。更为现实的定位是：把计算语言学看做理论语言学和计算机技术的桥梁，通过计算语言学家的工作来沟通语言学理论和计算机技术，来形成语言学技术（linguistic technology，如：针对某种语法体系的语法解释器和分析器，言语合成算法等），从而完成语言学理论在计算机上的应用。因为，在语言学理论和计算机处理技术之间存在着很深的鸿沟，一般的语言学理论研究的是抽象的语言能力(competence)，即理想的说话人和听话人的内在的语言知识；而不研究具体的语言运用(performance)，即语言知识在实际的语言活动中是怎样运用的。但是，计算机只能处理活动和过程性的知识。因此，计算语言学一直在尝试通过把语言学理论转变为算法（它能模拟遵守语言学理论和语言能力语法中所包含的各种语言学限制和概括的语言行为），来沟通语言能力语法和某种要适应用机器处理的特定的语言运用。（注：详见Halvorsen(1988)§2:The leap from linguistic theory to programs,pp.200-201。）事实上，更大的矛盾在于：语言学理论基本上是描述性的，而计算机技术中的算法描述和编程语言则基本上是过程性的。下面，我们简要地讨论这种矛盾及其解决办法。

一般地说，计算机要处理自然语言（最终目的是抓住句子的意义），首先必须对输入句进行句法分析(parsing)，从没有显性结构标记的符号串上找出结构来，即识别输入句的各个构成成分以及它们之间的关系，比如确定句子的主要动词及其主语和宾语，确定修饰成分及其中心语等。要分析句子的结构就需要语法的指导，正

是语法提供了一种语言的结构成分和符号串跟结构之间的关系的明确定义。在计算语言学上，通常称一个能根据一部特定的语法来分析句子（确定句子的推导过程）的程序为分析器(parser)。这种分析程序主要涉及两部分内容：(i)一组语法规则，它们由某种形式化的语法理论组织在一起，形成某种语法形式体系(grammaticalformalism)；(ii)一种控制机制(control mechanism)，它决定在分析过程中怎样运用语法规则、怎样保持对于各种业已发现的成分的记录、使程序在有限步运算后找出结构，即形成某种分析算法(parsing algorithms)。大家知道，程序是用编程语言(programming languages)编写的。而编程语言基本上是过程性的表示体系(procedural representation)，因为编程的目的本来就是给计算机提供一套明确而详尽的怎样干某事的指令(instructions)。但是，语法规则通常都是陈述性的(declarative)，而不是过程性的；它可以告诉我们一个句子往往由一个NP和一个VP构成，但并不告诉我们怎样用一个NP和一个VP去构成一个句子。面对这种语言学理论和计算机技术之间的不适配，有两种解决问题的思路：第一种，把陈述性的语法形式体系改变为过程性的语法形式体系，用过程性的形式体系来表示和组织语法规则。比如，利用转移网络这种形式机制的RTN语法(recursive transition network grammar)和ATN语法(augmented transition networkgrammar)就是一种过程性的语法体系。第二种思路是，把过程性的编程语言改变为陈述性的编程语言，用陈述性的表示体系（逻辑形式）来描述问题；只告诉机器要解决什么问题，但不说怎样去解决，让机器用定理证明的办法，通过自动推理去获得这方面的信息。Prolog就是这样一种基于逻辑推理的程序设计语言，这种逻辑程序设计语言(logic programming language)是一种陈述性（表示问题）语言，其控制（如何求解）过程由逻辑程序设计系统本身实现，无需程序设计人员给出解题算法。于是，为了充分利用这种编程语言的内在特性，基于Prolog的分析器应该把所要分析的问题看做是一个定理证明的问题。所有这类用便于Prolog编译的方式来表示语言学规则的语法形式体系，都叫做逻辑语法(logic grammar)。其中，限定子句语法(Definite Clause Grammar,DCG)就是一种逻辑语法。DCG是一种增强的上下文无关语法(Augmented Context-Free Grammar)，它的生成能力不低于ATN语法。更为重要的是，用限定子句表示的语法规则本身就是逻辑程序设计语言Prolog的可执行程序。换句话说，Prolog系统可以直接解释用DCG形式表示的语法规则，而无需像ATN那样另外再设计一个句法分析器（规则解释程序）来完成这个任务。

可见，计算机技术和语言学理论是相互影响、相互促进的。这造成了计算语言学和理论语言学的紧密合作，并且产生出丰硕的成果。比如，广义短语结构语法(Generalized PhraseStructure Grammar,GPSG)和词汇功能语法(Lexical Functional Grammar ,LFG)都是陈述性的语法形式体系，它们都受到M.Kay(1979)的计算语言学著作Unification Grammar（合一语法）的影响。其中，LFG是理论语言学家(J.Bresnan)和计算语言学家(R.Kaplan)的合作成果，GPSG的部分作者担任过大型的计算语言学项目的顾问。随着这种理论语言学和计算语言学的会聚(convergence)，也有许多计算语言学项目采用GPSG或LFG作为其语法形式体系，从而实现了从语言学理论到计算机技术的转变。（注：详见Halvorsen(1988)§4:Parsing,pp.204-210;Gazdar & Mellish(1987)§1.3:Towards Declarative Formalism,pp.228-229，§2:The Imposition ofStructure,pp.229-235；石纯一等(1993)§2.6：归结法和Prolog语言，第64-68页；第九章：句法分析，第333-422页。）

5.逻辑主义取向：着眼于语言学知识的自动发现

值得注意的是，最近出版的一些计算语言学著作，作者在计算语言学的定义中特意强调了语言的计算结构和计算模型。例如：

(1)计算语言学旨在以自然语言处理（包括理解、生成、人机对话、机器翻译以及语音／文字输入的后处理等）为技术背景，揭示自然语言的词法、句法、语义、语用诸平面及其相互作用的计算结构，把语言学知识重塑成可以转化为产品的计算模型。——白硕(1995)，第2页

(2)现代计算语言学是通过建立形式化的计算模型来分析、理解和处理语言的学科。……广义地讲，计算语言学是研究字符串的结构以及结构和意义的关系的学科。——翁富良　王野翊(1998)，第1、9页

按照白硕(1995)的理解，要建造一个处理自然语言的计算机系统，必须有大量的语言学知识作后盾；但是，语言学知识的发现工作主要是以手工的方式进行的。因此，利用计算机来自动（或辅助）发现语言学知识，将极大地提高研究的效率、扩大研究的规模，把语言学家从收例句、制卡片、画表格等烦琐的事务中解放出来。所谓语言学知识的发现，指的是从一个由例句组成的语料库中发现特定的自然语言规律。这种从一组事例中发现一般规律的认知活动，在逻辑上被描述成一种“归纳”过程。作者决心研究语言学规则这种特殊形式的知识的发现的逻辑实质，全面地展示跟语言学知识发现有关的各个层次上的形式化机制——从数学建模、逻辑分析、算法描述、具体实现直到结果的语言学解释。作者采用语言学中经典的分布分析的思想，并针对真实语料的各种特点，结合汉语的实际，从数学、逻辑、算法和实现各个角度，全面阐述了从语料中发现确定性语言学知识（主要是词类和句法规则）的理论和方法。这种计算语言学工作对语言学家来说是比较亲切的，因为它在相当程度上模拟了语言学家发现语言学规则的过程。

白硕(1995)的研究有着明显的逻辑主义追求，那就是通过研究语言学知识的发现来探索归纳法的逻辑机制和

计算结构。一般地说，从逻辑上看，人类的思维活动不外基于演绎法和基于归纳法两类。演绎法常常是从一些多少已经抽象化、形式化的前提出发，演绎出种种结论来。只要前提中含有可以互相消解(resolve)的对象，就一定可以衍生出新的命题来。显然，从前提演绎出结论是计算机可以胜任的工作。而归纳法常常是从未充分抽象化、形式化的大量个别事例出发，希望从中抽象出有用的概念、模式、定理来。这种工作能不能用计算机来完成呢？由于在使用归纳法的时候（比如，划分词类、发现句法模式等），目标的确立、是否达到目标的判别、达到目标的手段的建立等都是通过反复尝试而逐步建立起来的。对于这种缺少确定性的过程，计算机是很难单独完成的。怎么办呢？答案是建立一个人机共生的系统，由人来负责设定目标和手段、由机来负责实现这种手段而不管目标是什么。如果有了这样的人机共生系统，就可以大大地提高工作的效率和质量。要想做到这一点，就必须进一步研究归纳的手段和逻辑机理。而白硕(1995)主要是以语言学问题为背景，提出许多关于归纳的概念和方法作为人机共生系统的基础。（注：详见马希文为白硕《语言学知识的计算机辅助发现》所写的序，科学出版社1995年版，第ii-iii页。）他特别强调归纳的非单调性、可错性的特点：已经归纳出来的规则总有可能被后来的事实证明是不正确的、需要修改的，然而在没有遇到这样的事实时，这些规则又可以认为是近似正确的、不妨使用的。作者就是用这种允许某种“逻辑跳跃”来达到一些好的猜测的方法以发现词类和句法规则，并希望这种机制不仅仅局限于语言学知识的发现，希望这种研究对于探索知识发现的一般途径、对于认识归纳和类比的逻辑实质有所贡献。

从方法论和哲学背景上看，计算语言学研究有理性主义和经验主义两大分野。理性主义方法认为：人的很大一部分语言知识是与生俱来的，即是由遗传决定的。受Chomsky内在语言官能(innate language faculty)学说的影响，计算语言学界很多人信奉理性主义。他们秉承人工智能研究中的符号主义传统，通过人工汇编初始语言知识（主要表示成形式规则）和推理系统来建立处理自然语言的符号系统。这种系统通常根据一套规则或程序，将自然语言“理解”为某种符号结构；再通过某种规则，从组成该结构的符号的意义上推导出该结构的意义。在一个典型的自然语言处理系统中，句法分析器(parser)按照人所设定的自然语言的语法把输入句分析为句法结构（一种特定形式的符号结构），再根据一套语义规则把语法符号结构映射到语义符号结构（如：逻辑表示、语义网络、中间语言等）。由于自然语言处理系统中的规则集通常是先验的，即是由人设计好以后赋予机器的，因而，这是一种典型的理性主义的方法。经验主义方法认为：人的知识只有通过感官传入、再通过一些简单的联想(association)和泛化(generalization)的操作才能获得，人不可能天生拥有一套有关语言的原则和处理方法。表现在计算语言学中，许多研究尝试从大量的语言数据中获取语言的结构知识，从而开辟了基于语料库的计算语言学这种经验主义的研究方法。其中的神经网络方法秉承了人工智能研究中的连结主义传统，由机器通过学习给定的实例（训练数据）之间的输入—输出关系，来获得神经元（人工神经节点）之间的连结强度（strength，或称“权”weight），以反映从输入状态到输出状态之间的映射关系。其中的统计学方法试图建立统计性的语言处理模型，并由语料库中的训练数据来估计统计模型中的参数。比如，§2中介绍的词类的自动标注，其做法是先使用少量已经人工标注的语料进行训练，然后将学到的词类标记的共现概率分布用于标注尚未标注的文本。这都是通过学习训练实例来获得某种语言处理能力的，因而是典型的经验主义的研究方法。（注：详见翁富良、王野翊《计算语言学导论》§1.3：计算语言学研究的基本方法，中国社会科学出版社1998年版，第4-8页。）简而言之，理性主义强调基于规则的方法，经验主义强调基于学习的方法。而白硕(1995)的工作则是尝试兼采这两种方法之长又避免这两种方法之短。粗略地说，这是一种企图发现规则而不是赋予规则、基于语料库但不拘于统计学方法的路子。作者考虑到仅靠统计学方法是无法从语料中发现确定性的语言学规则的，因而尝试一种从精炼语料库中动态地归纳规则的方法。这种从语料库中通过学习来获得符号处理系统中的规则集的方法，在本质上是归纳逻辑。这种方法一方面用到符号处理系统中的规则表达，但规则又是从语料库中经验地获得的，因而，就其本性而言是一处经验主义的方法。（注：详见白硕(1995)§1.1，第1-5页；翁富良、王野翊(1998)§1.3，第4-8页。）

6.结语：并非悖论——用计算机和为计算机研究语言

最近几年，国际计算语言学界对计算语言学的定义逐步形成下面这种共识：计算语言学是用计算机和为计算机研究语言的学科。

说计算语言学的特点是用计算机来研究语言，这既有其通俗易懂的一面，又有其浅显误导的一面。其通俗性表现在：人们很容易想到计算语言学是把计算机作为工具来使用的，比如用计算机收集语料、分类整理、分布统计、提取各种数据等。这跟化学、物理学、生物学中的计算化学、计算物理学、计算生物学有点相近，它们或者运用简单的方程和算法在计算机上进行大量的重复运算，或者用计算机对实验结果进行十分精细的计算分析、反复提高以得到一种新的理论。其误导性表现在：人们只想到用计算机这种电子装置作为语言研究的工具，而忽略了用计算机科学的理论、概念和方法来研究语言这一点。我们认为这一点才是计算语言学更本质、更深刻的特

点。像§5介绍的白硕(1995)用理论计算机科学的观点剖析当代语言学的方法、并进行计算模拟的做法，在一定程度上展示了这类研究的理论魅力和实用价值。

为计算机研究语言，是指为了计算机能处理自然语言而研究语言。这包括两方面的工作：(1)对自然语言的结构和意义规律进行挖掘，提炼出便于形式化和算法化的句法、语义规则，建立合适的语法学理论模型，来更好地组织语言的句法、语义规则；(2)把语言学家对语言的句法、语义、语用诸平面上的研究成果进行数学概括，用某种形式化体系来组织和表示语言的结构和意义规则，再找出恰当的算法来描述句子的结构分析或语义解释的严格的步骤(procedure)，最后根据算法用相应的计算机语言来编程实现。

在为计算机研究语言这一点上，计算语言学有别于计算化学和计算神经科学。在计算化学中，并没有为计算机研究化学这种任务；在计算神经科学中，也没有为计算机研究神经的结构和功能这种任务。那么，为什么计算语言学要特别地强调为计算机研究语言这一点呢？原因可能有两点：(1)语言学的研究对象是自然语言，语言学的研究工具（用以描写语言现象、表述语言规律、总结研究结果）也是自然语言。也就是说，自然语言既是语言研究的对象语言，也是语言研究的元语言。由于计算机无法直接理解自然语言，因而首先必须把用自然语言表述的语言规律形式化、符号化。(2)语言是一种心智(mind)现象，是跟人的认知、心理密切相关的；为了让计算机能理解自然语言，必须以计算机为信息加工模型来考察人类语言理解的心理过程，以便在计算机上模拟实现。

可见，用计算机和为计算机研究语言并不是一种悖论，而是计算机语言学的本质特征。

【参考文献】

1.白硕(1995)《语言学知识的计算机辅助发现》，科学出版社。

2.范继淹、徐志敏(1980)《自然语言理解的理论和方法》，《国外语言学》第5期。

3.冯志伟(1992)《计算语言学对理论语言学的挑战》，《语言文字应用》第1期。

4.冯志伟(1996)《自然语言的计算机处理》，上海外语教育出版社。

5.桂诗春、宁春岩(1997)《语言学方法论》，外语教学与研究出版社。

6.黄昌宁(1990)《语料库语言学》，《中国计算机用户》第11期。

7.黄奕(1985)《认知过程的语言》，《国外语言学》第3期。

8.黄建烁(1991)《计算语言学研究综述》，《国际学术动态》第4期。

9.陆致极(1990)《计算语言学导论》，上海教育出版社。

10.马希文(1986)《计算机和思维科学》，见钱学森主编《关于思维科学》，人民出版社。

11.钱锋(1990)《计算语言学引论》，学林出版社。

12.沈政、林庶之(1992)《脑模拟和神经计算机》，北京大学出版社。

13.石纯一、黄昌宁、王家@①(1993)《人工智能原理》，清华大学出版社。

14.翁富良、王野翊(1998)《计算语言学导论》，中国社会科学出版社。

15.杨抒(1988)《自然语言的认知模型》，《计算机科学》第3期。

16.袁毓林(1996)《语言的认知研究和计算分析》，删节本见《语言文学应用》第1期。全文见罗振声、袁毓林主编《计算机时代的汉语和汉字研究》，清华大学出版社。

17.Gazdar,G.& Mellish,C.(1987)Computational linguistics,in J.Lyons,etc.(ed.)New Horizons in Linguistics 2.Penguin Books.

18.Grishman,Ralph(1986)Computational Linguistics:An Introduction.Cambridge University Press.

19.Halvorsen,Per-Kristian(1988)Computer applications of linguistic theory in F.J.Newmeyer(ed.) Linguistics:The Cambridge Survey,Vol.Ⅱ,Linguistic Theory:Extentions and Implications.Cambridge University Press.

20.Winograd,Terry(1983)Languageasa Cognitive Process.Addison-Wesley Publishing Company,Inc.中文简介请看黄奕(1985)。

字库未存字注释：

@①广内加钦

文章来源：人大复印资料

• 上一条：已经没有了
• 下一条：儿童语言障碍的语言学研究 (7-7)

相关专题：无	相关信息：-
尚无信息	• 2007-2008年度国家文化出口重点项目目 (5-7) • "一语双文"的理论基础和面临的困难--简 (5-1) • IT-常用词汇 (3-18) • 四大名著的外文译名 (1-20) >>更多